

## Research Article

# Automated Design of an FDI System for the Wind Turbine Benchmark

**Carl Svärd<sup>1,2</sup> and Mattias Nyberg<sup>1,2</sup>**

<sup>1</sup> Department of Electrical Engineering, Linköping University, 58183 Linköping, Sweden

<sup>2</sup> Scania CV AB, 15187 Södertälje, Sweden

Correspondence should be addressed to Carl Svärd, carl@isy.liu.se

Received 1 July 2011; Accepted 4 October 2011

Academic Editor: Jakob Stoustrup

Copyright © 2012 C. Svärd and M. Nyberg. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an FDI system for the wind turbine benchmark designed by the application of a generic automated method. No specific adaptation of the method for the wind turbine benchmark is needed, and the number of required human decisions, assumptions, as well as parameter choices is minimized. The method contains in essence three steps: generation of candidate residual generators, residual generator selection, and diagnostic test construction. The proposed FDI system performs well in spite of no specific adaptation or tuning to the benchmark. All faults in the predefined test sequence can be detected and all faults, except a double fault, can also be isolated shortly thereafter. In addition, there are no false or missed detections.

## 1. Introduction

Wind turbines stand for a growing part of power production. The demands for reliability are high, since wind turbines are expensive and their off-time should be minimized. One potential way to meet the reliability demands is to adopt fault tolerant control (FTC), that is, prevent faults from developing into failures by taking appropriate actions. A typical action is reconfiguration of the control system. An essential part of an FTC system is the fault detection and isolation (FDI) system, see, for example, [1]. To obtain good detection and isolation of faults, model-based FDI is often necessary.

Design of a complete model-based FDI-system is a complex task and involves by necessity several decisions, for example, method choices, tuning of parameters, and assumptions regarding noise distributions and the nature of the faults to be diagnosed. In general, an optimal solution requires detailed knowledge of the behavior of the considered system, something that is rarely available for real applications. In this paper, inspired by the work with real-industrial applications, we propose an automated design method that minimizes the number of required human decisions and assumptions. Furthermore, we investigate the potential of

designing an FDI system for the wind turbine benchmark, see [2], using this automated method.

The design method is composed of three main steps. In the first step, a large set of candidate residual generators are generated using the algorithm described in [3]. In the second step, the residual generators most suitable to be included in the final FDI system are selected and realized by means of a greedy selection algorithm, based on ideas elaborated in [4]. The realization, or construction, of residual generators is done by the use of the algorithms presented in [5]. In the third and final step, we design diagnostic tests based on the residuals obtained as output from the selected set of residual generators. The diagnostic test relies on a novel methodology based on a comparison of the probability distributions of no-fault residuals, estimated offline using no-fault training data, and the distributions of residuals estimated online using current data.

As it turns out, the proposed FDI system performs well when evaluated on the test sequence described in [2]. A tailor-made FDI system perfectly tuned for the wind turbine benchmark would probably perform better than the one we propose. However, in relation to the minimal effort required for application of the automated design method, and in spite of no extra tuning or specific adaptation to the benchmark,

the performance of the FDI system is satisfactory; all faults in the test sequence can be detected within feasible time, and there are no false or missed detections. Further, all faults, except a double fault, can also be isolated.

The wind turbine benchmark model and the strategy used for modeling of faults are described in Section 2. Section 3 presents an overview of the design method. The method for constructing residual generators is described in Section 4, and the approach used for selecting residual generators is described in Section 5. The method for design of diagnostic tests, and the fault isolation scheme is considered in Section 6. Some implementation-specific details are discussed in Section 7. The performance of the designed FDI system is evaluated and discussed in Section 8, Section 9 concludes the paper.

## 2. The Wind Turbine Model

The wind turbine system is described and modeled in [2], to which is referred for details. The considered wind turbine system has three rotor blades and the system contains four subsystems: blade and pitch system, drive train, generator and converter, and controller, see Figure 1 and Table 1.

**2.1. State-Space Realization of Transfer Functions.** The pitch system and converter are modeled as frequency domain transfer functions. The residual generation algorithm we intend to apply, assumes a model described in differential and algebraic equations. To obtain a model in this form, the transfer functions are realized as time-domain state-space systems.

The relation between pitch angle reference  $\beta_r$  and pitch angle output  $\beta_i$ , for each of the three blades and thus for  $i = 1, 2, 3$ , can be realized in state-space form using observable canonical form, see, for example, [6], as follows:

$$\dot{x}_{\beta_{i1}}(t) = -2\zeta\omega_n x_{\beta_{i1}}(t) + x_{\beta_{i2}}(t), \quad (1a)$$

$$\dot{x}_{\beta_{i2}}(t) = -\omega_n^2 x_{\beta_{i1}}(t) + \omega_n^2 \beta_r(t), \quad (1b)$$

$$\beta_i(t) = x_{\beta_{i1}}(t), \quad (1c)$$

where  $\zeta$ ,  $\omega_n$  are parameters, and  $x_{\beta_{i1}}$ ,  $x_{\beta_{i2}}$ , state variables. Using the same approach, the relation between converter reference  $\tau_{g,r}$  and output  $\tau_g$  can be written as

$$\dot{x}_{\tau_g}(t) = -\alpha_{gc} x_{\tau_g}(t) + \alpha_{gc} \tau_{g,r}(t), \quad (2a)$$

$$\tau_g(t) = x_{\tau_g}(t), \quad (2b)$$

where  $\alpha_{gc}$  is a parameter, and  $x_{\tau_g}$  is the state variable.

**2.2. Fault Modeling.** The set of faults to consider for the wind turbine is specified in [2] and given by

$$F = \left\{ \Delta\beta_1, \Delta\beta_2, \Delta\beta_3, \Delta\tau_g, \Delta\omega_g, \Delta\beta_{1,m1}, \Delta\beta_{1,m2}, \Delta\beta_{2,m1}, \Delta\beta_{2,m2}, \Delta\beta_{3,m1}, \Delta\beta_{3,m2}, \Delta\omega_{r,m1}, \Delta\omega_{r,m2}, \Delta\omega_{g,m1}, \Delta\omega_{g,m2} \right\}, \quad (3)$$

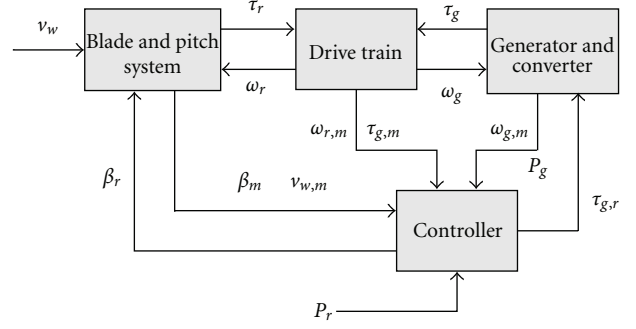


FIGURE 1: Overview of the wind turbine system.

TABLE 1: Signals in the wind turbine system.

Signal	Description
$v_w$	Wind speed
$v_{w,m}$	Wind speed measurement
$\beta_r$	Pitch angle reference
$\beta_m$	Pitch angle measurement
$\omega_r$	Angular rotor speed
$\omega_{r,m}$	Angular rotor speed measurement
$\omega_g$	Generator rotor speed
$\omega_{g,m}$	Generator rotor speed measurement
$\tau_r$	Rotor torque
$\tau_g$	Generator torque
$\tau_{g,r}$	Generator torque reference
$\tau_{g,m}$	Generator torque measurement
$P_r$	Power reference
$P_g$	Generator power

where  $\Delta\beta_1$ ,  $\Delta\beta_2$ ,  $\Delta\beta_3$ , and  $\Delta\tau_g$  are actuator faults,  $\Delta\omega_g$  is a system fault, and  $\Delta\beta_{1,m1}$ ,  $\Delta\beta_{1,m2}$ ,  $\Delta\beta_{2,m1}$ ,  $\Delta\beta_{2,m2}$ ,  $\Delta\beta_{3,m1}$ ,  $\Delta\beta_{3,m2}$ ,  $\Delta\omega_{r,m1}$ ,  $\Delta\omega_{r,m2}$ ,  $\Delta\omega_{g,m1}$ , and  $\Delta\omega_{g,m2}$  are sensor faults.

To incorporate fault information in the nominal model, we have chosen to model all faults as additive signals in corresponding equations. Thus, we are not taking into account all information regarding the nature of faults given in [2]. Consider, for example, fault  $\Delta\beta_1$  which represents an actuator fault in pitch system 1, see (1a)–(1c), resulting in changed dynamics of  $\beta_1$  due to dropped main line pressure or high air content in the oil. One possible way to model this fault would be as a deviation in parameters  $\omega_n$  and  $\zeta$  in (1a) and (1b). With the chosen approach, the fault is instead modeled as an additive signal in (1c) for  $i = 1$ , that is,  $\beta_1 = x_{\beta_{11}} + \Delta\beta_1$ .

Note that the adopted fault modeling approach is general and no assumptions are made regarding, for example, the time-behavior of faults. Thus, the approach is able to handle, for example, multiplicative faults even though the fault signal is assumed to be additive. Consider, for example, a multiplicative fault in  $\beta_1$  given by  $\beta_1 = \delta \cdot x_{\beta_{11}}$ , where  $\delta \neq 1$ , which can be equivalently described by  $\beta_1 = x_{\beta_{11}} + \Delta\beta_1$ , where  $\Delta\beta_1 = x_{\beta_{11}}(\delta - 1)$ .

The main argument for using this, more general, approach is that we consider it hard, or even impossible, to know exactly how a faulty component behaves in reality. Furthermore, data from all fault cases for evaluation and validation of a more-detailed model are seldom available. Modeling faults in this way also results in a minimum of fault modes. This is beneficial since it gives a smaller model which simplifies several steps in model-based diagnosis, for example, residual generation and isolation. In addition, regarding how diagnosis information is utilized, for example, for fault tolerant control, it is unnecessary to distinguish between different fault modes if they are associated with the same action or consequence. Indeed, this applies to all sensor faults in the wind turbine, since the system should be reconfigured regardless of the type of sensor fault, that is, *fixed value* or *gain factor*, see [2, Table 2]. Last, but not least, an additional important motivator is simplicity, since extending the nominal model with additive fault signals in this way is straightforward and easy.

**2.3. Model Extensions.** According to [2], the same pitch angle reference signal  $\beta_r$  is fed to all three pitch systems (1a)–(1c), that is,  $\beta_{i,r} = \beta_r$  for  $i = 1, 2, 3$ . However, according to the provided Simulink model, see [7], the individual reference signals are instead calculated in a control loop outside the pitch system as

$$\beta_{i,r} = \beta_r + \beta_i - \left( \frac{\beta_{i,m1} + \beta_{i,m2}}{2} \right), \quad i = 1, 2, 3, \quad (4)$$

where  $\beta_i$  is given by (1a)–(1c), and  $\beta_{i,m1}$  and  $\beta_{i,m2}$  are sensor measurements. To incorporate this information in the design of the FDI system, the original wind turbine model is extended with the relations between  $\beta_{i,r}$  and  $\beta_r$  given by (4).

**2.4. The Model with Faults.** The complete model of the wind turbine model, with fault signals denoted by  $\Delta$ , used in this work for design of an FDI system is given below:

$$\begin{aligned} e_1 : \tau_r &= \sum_{i=1}^3 \frac{\rho \pi R^3 C_q(\lambda, \beta_i) v_w^2}{6}, \\ e_2 : \lambda &= \frac{\omega_r R}{v_w}, \\ e_3, e_5, e_7 : \dot{x}_{\beta_{i1}} &= -2\zeta \omega_n x_{\beta_{i1}} + x_{\beta_{i2}}, \quad i = 1, 2, 3, \\ e_4, e_6, e_8 : \dot{x}_{\beta_{i2}} &= -\omega_n^2 x_{\beta_{i1}} + \omega_n^2 \beta_{i,r}, \quad i = 1, 2, 3, \\ e_9, e_{10}, e_{11} : \beta_i &= x_{\beta_{i1}} + \Delta \beta_i, \quad i = 1, 2, 3, \\ e_{12} : \dot{\omega}_g &= \left( \frac{\eta_{dt} B_{dt}}{N_g J_g} \right) \omega_r + \left( \frac{-(\eta_{dt} B_{dt} / N_g^2) - B_g}{J_g} \right) \omega_g \\ &\quad + \left( \frac{\eta_{dt} K_{dt}}{N_g J_g} \right) \theta_\Delta - \left( \frac{1}{J_g} \right) \tau_g + \Delta \omega_g, \end{aligned}$$

$$\begin{aligned} e_{13} : \dot{\omega}_r &= -\left( \frac{B_{dt} - B_r}{J_r} \right) \omega_r + \left( \frac{B_{dt}}{N_g J_r} \right) \omega_g \\ &\quad - \left( \frac{K_{dt}}{J_r} \right) \theta_\Delta + \left( \frac{1}{J_r} \right) \tau_r, \\ e_{14} : \dot{\theta}_\Delta &= \omega_r - \left( \frac{1}{N_g} \right) \omega_g, \\ e_{15} : \dot{x}_{\tau_g} &= -\alpha_{gc} x_{\tau_g} + \alpha_{gc} \tau_{g,r}, \\ e_{16} : \tau_g &= x_{\tau_g} + \Delta \tau_g, \\ e_{17} : P_g &= \eta_{gc} \omega_g \tau_g, \\ e_{18}, e_{20}, e_{22} : \beta_{i,m1} &= \beta_i + \Delta \beta_{i,m1}, \quad i = 1, 2, 3, \\ e_{19}, e_{21}, e_{23} : \beta_{i,m2} &= \beta_i + \Delta \beta_{i,m2}, \quad i = 1, 2, 3, \\ e_{24}, e_{25} : \omega_{r,mj} &= \omega_r + \Delta \omega_{r,mj}, \quad j = 1, 2, \\ e_{26}, e_{27} : \omega_{g,mj} &= \omega_g + \Delta \omega_{g,mj}, \quad j = 1, 2, \\ e_{28} : v_{w,m} &= v_w, \\ e_{29} : \tau_{g,m} &= \tau_g, \\ e_{30} : P_{g,m} &= P_g, \\ e_{31}, e_{32}, e_{33} : \beta_{i,r} &= \beta_r + \beta_i - \left( \frac{\beta_{i,m1} + \beta_{i,m2}}{2} \right), \quad i = 1, 2, 3. \end{aligned} \quad (5)$$

### 3. Overview of Design Method

The proposed FDI system for the wind turbine is comprised of three subsystems: residual generation, fault detection, and fault isolation, see Figure 2.

Measurements, that is, sensor readings, from the wind turbine are fed to a bank of residual generators whose output is a set of residuals. The residuals are used as input to the fault detection block, which contains diagnostic tests based on the residuals. The output from this block, one signal for each residual, indicates if a fault has been detected in the part of the system monitored by the corresponding residual. The result from the fault detection is fed to the fault isolation block in which the detected fault(s) are isolated.

The proposed method supports design of the residual generation and fault detection blocks. Design of the fault isolation block is briefly discussed in Section 6.2. The method contains three essential steps:

- (1) generate candidate residual generators,
- (2) select and realize residual generators,
- (3) construct diagnostic tests,

see Figure 3. In the first step, a large set of candidate residual generators are generated. In the second step, the residual generators most suitable to be included in the final FDI system are selected and realized. In the third and final step,

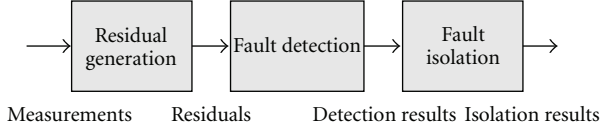


FIGURE 2: Schematic overview of the FDI system.

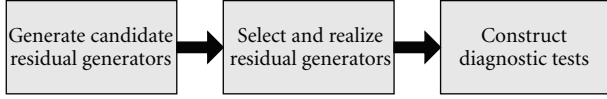


FIGURE 3: Overview of the design method.

we design diagnostic tests based on the residuals obtained as output from the selected set of residual generators.

In the subsequent sections, we describe in detail the different steps of the design method used to create the proposed FDI system for the wind turbine benchmark system. As input to the design method, or prerequisites, we assume a model of the system and no-fault training data. The data is assumed to be expressed as measurements, either real or simulated, of the inputs and outputs of the model in realistic and representative no-fault operating conditions.

## 4. Residual Generation

The set of residual generators used in the FDI system are based upon the ideas originally described in [8], where unknown variables in a model are computed by solving equation sets one at a time in a sequence and a residual is obtained by evaluating a redundant equation. Similar approaches are described and exploited in, for example, [1, 5, 9–13]. This class of residual generation methods, referred to as *sequential residual generation*, has shown to be successful for real applications and also has the potential to be automated to a high extent.

**4.1. Sequential Residual Generation.** Some concepts and results of sequential residual generation given in [5], to which we also refer for technical details, will now be briefly recapitulated. We consider a model  $(E, X, D, Y)$  to be a set of differential and algebraic equations  $E = \{e_1, e_2, \dots, e_{n_E}\}$  containing unknown variables  $X = \{x_1, x_2, \dots, x_{n_X}\}$ , differential variables  $D = \{\dot{x}_1, \dot{x}_2, \dots, \dot{x}_{n_X}\}$ , and known variables  $Y = \{y_1, y_2, \dots, y_{n_Y}\}$ . The equations in  $E$  are, without loss of generality, assumed to be on the form

$$e_i : f_i(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}) = 0, \quad i = 1, 2, \dots, n_E, \quad (6)$$

where  $\dot{\mathbf{x}}$ ,  $\mathbf{x}$  and  $\mathbf{y}$ , are vectors of the variables in  $D$ ,  $X$ , and  $Y$ , respectively. Note that the model of the wind turbine presented in Section 2.4 can trivially be cast into this form.

**4.1.1. Computation Sequence.** As said above, the main idea in sequential residual generation is to compute unknown variables in the model by solving equation sets one at a time in a sequence and then evaluate a redundant equation to

obtain a residual. An essential component in the design of a residual generator is therefore a computation sequence, which describes the order in which the variables should be computed. In [5], a computation sequence is defined as an ordered set of variable and equation pairs:

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \dots, (V_k, E_k)), \quad (7)$$

where  $V_i \subseteq X \cup D$  and  $E_i \subseteq E$ . The computation sequence  $\mathcal{C}$  implies that first the variables in  $V_1$  are computed from equations  $E_1$ , then the variables in  $V_2$  from equations  $E_2$ , possibly using the already computed variables in  $V_1$ , and so forth.

For an example, consider the computation sequence:

$$\mathcal{C} = \left( (\{\tau_g\}, \{e_{29}\}), (\{\omega_r\}, \{e_{24}\}), (\{\dot{\theta}_\Delta\}, \{e_{14}\}), (\{\dot{\omega}_g\}, \{e_{12}\}) \right), \quad (8)$$

for computation of a subset of the unknown variables in wind turbine model presented in Section 2.4. According to the computation sequence (8), the series of computations begins with computation of variable  $\tau_g$  using equation  $e_{29}$ , then variable  $\omega_r$  is computed using equation  $e_{24}$ , and so on, ending with computation of variable  $\dot{\omega}_g$ , or in fact  $\omega_g$  from equation  $e_{12}$ .

By construction, see [5], it is guaranteed that no variable is needed before it has been computed. Hence, the series of computations described by the computation sequence exhibit an upper triangular structure. For the computation sequence (8), this series of computations is given by

$$\tau_g = \tau_{g,m}, \quad (9a)$$

$$\omega_r = \omega_{r,m1}, \quad (9b)$$

$$\dot{\theta}_\Delta = \omega_r - \left( \frac{1}{N_g} \right) \omega_g, \quad (9c)$$

$$\begin{aligned} \dot{\omega}_g = & \left( \frac{\eta_{dt} B_{dt}}{N_g J_g} \right) \omega_r + \left( \frac{-\left( \eta_{dt} B_{dt} / N_g^2 \right) - B_g}{J_g} \right) \omega_g \\ & + \left( \frac{\eta_{dt} K_{dt}}{N_g J_g} \right) \theta_\Delta - \left( \frac{1}{J_g} \right) \tau_g. \end{aligned} \quad (9d)$$

Whether it is possible or not to compute the specified variables from the corresponding equations depends naturally on the properties of the equations. Equally important are, however, prerequisites in terms of *causality assumption*, that is, regarding integral and/or derivative causality, and the properties of the *computational tools*, that are available for use, for a detailed discussion, see, for example, [5]. The computation sequence (8) makes use of solely integral causality when the variables  $\theta_\Delta$  and  $\omega_g$  are computed using equations  $e_{14}$  and  $e_{12}$ , respectively.

**4.1.2. Sequential Residual Generator.** Having computed the unknown variables in  $V_1 \cup V_2 \cup \dots \cup V_k$ , according to the

computation sequence  $\mathcal{C}$  in (7), a residual can be obtained by evaluating a redundant equation  $e$ , that is,  $e \in E \setminus E_1 \cup E_2 \cdots \cup E_k$  with  $\text{var}_X(e) \subseteq \text{var}_X(E_1 \cup E_2 \cdots \cup E_k)$ , where the operator  $\text{var}_X(\cdot)$  returns the unknown variables that are contained in an equation set. A residual generator based on a computation sequence  $\mathcal{C}$  and redundant equation  $e$  is referred to as a *sequential residual generator*.

The computation sequence (8) together with equation  $e_{26}$  constitutes a sequential residual generator for the wind turbine model. When all variables in the computation sequence (8) have been computed according to (9a)–(9d), the residual is computed as  $r = \omega_{g,m1} - \omega_g$ .

**4.1.3. Finding Sequential Residual Generators.** Regarding implementation aspects, for example, complexity and computational load, it is unnecessary to compute variables that are not contained in the residual equation, or not used to compute any of the variables contained in the residual equation. Furthermore, it is also desirable that computation of variables in each step is performed from as small equation sets as possible. It can be shown, see [5], that the equations in a computation sequence fulfilling the above properties, together with a redundant residual equation, in fact correspond to a minimal structurally overdetermined (MSO) set, see [3]. In other words, a necessary condition for the existence of a sequential residual generator for a model is that the model, or submodel, is an MSO set.

**4.2. Candidate Residual Generators.** As indicated above, a first step when searching for a sequential residual generator for a model may be to find an MSO set in the model. Thus, an MSO set can be regarded as a *candidate residual generator*. There are efficient algorithms for finding all MSO sets in large equation sets, see, for example, [3].

Consider now the model of the wind turbine described in Section 2.4, with equations  $E = \{e_1, e_2, \dots, e_{33}\}$ , unknown variables:

$$X = \left\{ \tau_r, \beta_1, \lambda, v_w, \beta_2, \beta_3, \omega_r, x_{\beta_{11}}, x_{\beta_{12}}, \beta_{1,r}, x_{\beta_{21}}, x_{\beta_{22}}, \beta_{2,r}, x_{\beta_{31}}, x_{\beta_{32}}, \beta_{3,r}, \omega_g, \theta_\Delta, \tau_g, x_{\tau_g}, P_g \right\}, \quad (10)$$

and known, that is, measured, variables:

$$Y = \left\{ \beta_r, \tau_{g,r}, \beta_{1,m1}, \beta_{1,m2}, \beta_{2,m1}, \beta_{2,m2}, \beta_{3,m1}, \beta_{3,m2}, \omega_{r,m1}, \omega_{r,m2}, \omega_{g,m1}, \omega_{g,m2}, v_{w,m}, \tau_{g,m}, P_{g,m} \right\}. \quad (11)$$

In summary, the model contains 33 equations, 21 unknown variables, and 15 known variables. By utilizing the *structure*, that is, which unknown variables are contained in which equation, see, for example, [1], and a MATLAB implementation of the algorithm presented in [3], 1058 MSO sets were found in total.

## 5. Selecting Residual Generators

It is not feasible to implement and use all 1058 candidate residual generators, that is, MSO sets, in the final FDI system.

A more attractive approach is instead to pick, from the set of all candidate residual generators, a smaller set of residual generators with desired properties.

**5.1. Desired Properties of Residual Generators.** The desired properties of the sought set of residual generators are as follows:

- (1) the set of residual generators should enable us to isolate all single faults from each other;
- (2) a set of residual generators of smaller cardinality is preferred before a larger one, given that the two sets have equal isolability properties;
- (3) a residual generator based on an MSO set of smaller cardinality is preferred before a residual generator based on an MSO set of larger cardinality, given that the two sets have equal detectability and isolability properties.

Properties 2 and 3 are mainly motivated by implementation aspects such as complexity, computational load, and numerical issues.

We will base the selection of residual generators on quantitative, structural properties of the MSO sets instead of more qualitative or analytical properties on the actual residual generators. The latter may result in better isolation performance but is considered intractable since it requires that residual generators are implemented, executed, and evaluated, and also access to representative measurement data for all fault cases.

**5.2. Fault Detectability and Isolability.** To be able to formally state the selection problem, the notions of detectability and isolability are needed. Assuming that each fault occurs in only one equation, let  $e_{f_i}$  denote the equation in an equation set  $E$  containing fault  $f_i$ , for example,  $e_{\Delta\beta_{1,m1}} = e_{18}$ , see Section 2. Note that if a fault  $f_j$  occurs in more than one equation, the fault  $f_j$  can be replaced with a new variable  $x_{f_j}$  in these equations, and the equation  $x_{f_j} = f_j$  added to the equation set. This added equation will then be the only equation where  $f_j$  occurs. To proceed, let  $(\cdot)^+$  denote an operator extracting the overdetermined part of a set of equations. According to [14], a fault  $f_i$  is *structurally detectable* in the equation set  $E$  if  $e_{f_i} \in (E)^+$  and *structurally isolable* from fault  $f_j$  in the equation set  $E$  if  $e_{f_i} \in (E)^+$  and  $e_{f_j} \notin (E)^+$ .

For an example, consider the equation set  $M = \{e_{26}, e_{29}, e_{24}, e_{14}, e_{12}\}$  containing the residual equation and equations from the computation sequence (7), studied in Section 4.1.1. First, we note that the equation set  $M$  is an MSO set due to the property of sequential residual generators mentioned in Section 4.1.3. Further, since  $M$  is an MSO set, it holds that  $(M)^+ = M$ , see, for example, [3]. Thus, it can for instance be deduced that fault  $\Delta\omega_g$  is structurally isolable from fault  $\Delta\beta_{1,m1}$  in  $M$ , since  $e_{\Delta\omega_g} = e_{12}$ ,  $e_{\Delta\beta_{1,m1}} = e_{18}$ , and it holds that  $e_{12} \in M$  and  $e_{18} \notin M$ , see Section 2.4.

By again utilizing the structure of the wind turbine model, the structural isolability properties of the model were



calculated. All considered faults, see Section 2.2, can be (structurally) isolated from each other in the wind turbine model.

**5.3. Selection Problem Formulation.** We will now formulate the selection problem in terms of properties on a set of MSO sets. To this end, let  $\mathcal{M}$  denote the set of all MSO sets in the model, and  $F$  the set of considered faults. Let  $f_i, f_j \in F$  and define the *isolation class* for  $(f_i, f_j)$  as

$$I_{f_i, f_j} = \{M \in \mathcal{M} : e_{f_i} \in (M)^+ \wedge e_{f_j} \notin (M)^+\}, \quad (12)$$

that is,  $I_{f_i, f_j}$  contains the MSO sets in  $\mathcal{M}$  in which fault  $f_i$  is structurally isolable from fault  $f_j$ . Further, let

$$\mathcal{I} = \{I_{f_i, f_j} : \forall (f_i, f_j) \in F \times F, f_i \neq f_j\} \quad (13)$$

denote the set of all isolation classes needed for full isolation of all faults in  $F$ . For the wind turbine benchmark model and the set of 15 faults considered in Section 2.2, the set  $\mathcal{I}$  contains in total  $15 \times 15 - 15 = 210$  isolation classes for single fault isolation of all 15 faults, that is,  $|\mathcal{I}| = 210$ , where the operator  $|\cdot|$  returns the cardinality of a set.

To be able to satisfy the isolability property 1 stated above, we want to find a set  $\mathcal{S} \subseteq \mathcal{M}$  with a nonempty intersection with all isolation classes, that is,

$$\forall I_{f_i, f_j} \in \mathcal{I}, \quad \mathcal{S} \cap I_{f_i, f_j} \neq \emptyset. \quad (14)$$

The property (14) on  $\mathcal{S}$  implies that we should find a so-called *hitting set* for  $\mathcal{I}$ . To satisfy the property 2, we want to find an  $\mathcal{S}$  so that  $|\mathcal{S}|$  is minimized. Thus, the sought hitting set for  $\mathcal{I}$  should be of minimal cardinality and we should find a so-called *minimal cardinality hitting set* (MHS) for  $\mathcal{I}$ .

There are several possibilities for a metric that helps us find an  $\mathcal{S}$  that satisfies property 3. We opt for simplicity and have, therefore, chosen to minimize  $\sum_{M \in \mathcal{S}} |M|$ . As an additional requirement, on top of 1, 2, and 3 in Section 5.1 we require that at least one residual generator can be constructed from every  $M \in \mathcal{S}$ .

**5.4. Solving the Selection Problem.** The problem of finding a minimal cardinality hitting set is known to be NP-hard, see, for example, [15]. To overcome the complexity issues, we have chosen to compute an approximate solution to the problem in an iterative manner with a greedy selection approach as elaborated in [4].

To accomplish this, we need to specify a *utility function*, that is, a function that evaluates the usefulness of a given MSO set, and also state the properties of a complete solution to the selection problem. Following the greedy selection approach, we add to the solution the MSO set with the largest utility until the solution is complete. Furthermore, we only add MSO sets from which at least one residual generator can be constructed.

**5.4.1. Characterization of a Solution.** We will now characterize a complete solution to the selection problem for use in

the selection algorithm. First, we define the *isolation class coverage* of a set of MSO sets  $\mathcal{S} \subseteq \mathcal{M}$  as

$$\sigma(\mathcal{S}) = \{I_{f_i, f_j} \in \mathcal{I} : \exists M \in \mathcal{S}, M \in I_{f_i, f_j}\}, \quad (15)$$

which states which of the isolation classes in  $\mathcal{I}$  that are covered by the MSO sets in  $\mathcal{S}$ . The property 1 in Section 5.1, that is, the isolation or hitting set property, can with the isolation class coverage notion be formulated as  $\sigma(\mathcal{S}) = \mathcal{I}$ . This characterizes a complete solution of the selection problem.

**5.4.2. Utility Function.** To evaluate a specific MSO set, we want to take into account the properties 1, 2, and 3, above. For a given MSO set  $M$ , we will use the utility function:

$$\mu_{\mathcal{I}}(M) = \gamma \left( \frac{|\sigma(\{M\})|}{|\mathcal{I}|} \right) + (1 - \gamma) \left( 1 - \frac{|M|}{|\widehat{M}|} \right), \quad (16)$$

where  $\widehat{M}$  is the MSO set in  $\mathcal{M}$  with the largest cardinality, and  $\gamma, 0 \leq \gamma \leq 1$ , a weighting factor. The term  $|\sigma(\{M\})|/|\mathcal{I}|$  in (16) tells how many of the isolation classes in  $\mathcal{I}$  are covered by the MSO set  $M$ . Since we aim at covering all isolation classes with a minimum of MSO sets, property 2, we want to pick an MSO set that maximizes this term. The term  $1 - (|M|/|\widehat{M}|)$  relates the cardinality of  $M$  to the cardinality of all other sets in  $\mathcal{M}$ . Picking an MSO set that maximizes this term in (16) hence corresponds to picking the MSO set with the smallest cardinality in  $\mathcal{M}$ . This will help us satisfy property 3. The weighting factor  $\gamma$  is used to trade between the two properties reflected by these two terms.

Note that an MSO set maximizing one term in (16) may minimize the other since an MSO set of larger cardinality likely covers more isolation classes than an MSO set of smaller cardinality.

**5.5. The Selection Algorithm.** The function `SELECTRESIDUALGENERATORS` used for selecting residual generators by means of greedy selection is given in Algorithm 1. Input to the function is a set of MSO sets  $\mathcal{M}$ , that is, a set of candidate residual generators, and a set of isolation classes  $\mathcal{I}$ . The output is a set of MSO sets  $\mathcal{S} \subseteq \mathcal{M}$  and a set of residual generators  $\mathcal{G}$  based on  $\mathcal{S}$ . The function `FINDCOMPUTATIONSEQUENCE`, described in [5], is used to find a computation sequence in accordance with Section 4.1, given a just-determined set of equations. The function `FINDCOMPUTATIONSEQUENCE` can be found in Algorithm 2.

For a formal discussion regarding the qualification of using a greedy heuristic for solving the residual generation selection problem, as well as the complexity properties of such algorithms, please refer to [4] and references therein.

**5.5.1. Selecting Residual Equation.** Note that the total number of sequential residual generators that potentially can be constructed from an MSO set equals the number of equations in the set. All residual generators created from the same MSO set, however, have equal fault detectability and isolability properties according to Section 5.2. Nevertheless,

```

function SELECTRESIDUALGENERATORS( $\mathcal{M}, \mathcal{I}$ )
 $\mathcal{S} := \emptyset$ 
 $\mathcal{G} := \emptyset$ 
while  $\mathcal{I} \neq \emptyset$  do
   $M := \arg \max_{M \in \mathcal{M}} \mu_{\mathcal{I}}(M)$ 
   $x := \text{var}_X(M)$ 
   $G := \emptyset$ 
  for all  $e \in M$  do
     $M' := M \setminus \{e\}$ 
     $\mathcal{C} := \text{FINDCOMPUTATIONSEQUENCE}(M', x)$ 
    if  $\mathcal{C} \neq \emptyset$  then
       $G := G \cup \{(\mathcal{C}, e)\}$ 
    end if
  end for
  if  $G \neq \emptyset$  then
     $\mathcal{S} := \mathcal{S} \cup \{M\}$ 
     $\mathcal{G} := \mathcal{G} \cup \{G\}$ 
  end if
   $\mathcal{M} := \mathcal{M} \setminus \{M\}$ 
   $\mathcal{I} := \mathcal{I} \setminus \sigma(\{M\})$ 
end while
return ( $\mathcal{S}, \mathcal{G}$ )
end function

```

ALGORITHM 1: Greedy selection of residual generators.

their actual fault detectability and isolability may differ due, for example, different sensitivity for noise, and so forth. To make the final selection of which of the residual generators created from an MSO set that should be included in the final diagnosis system, evaluation by means of execution using real measurements from different fault cases is needed. Since we in this work only assume that no-fault data is available, see Section 3, this is not possible.

In this work, the selection of which residual generator to create from a given MSO set is done so that the final deployment of the FDI system becomes as simple as possible. First of all, FINDCOMPUTATIONSEQUENCE was configured to prefer algebraic equations as residuals before differential equations, if possible. Second, in order to avoid implementation issues related to numerical differentiation, FINDCOMPUTATIONSEQUENCE was configured to prefer computation sequences using integral causality. Using this two-step heuristic, the selection of which residual generator to create from an MSO set, in practice, is more or less unambiguous. In those few cases where more than one candidate remains, we make an arbitrary selection.

**5.6. Selected Residual Generators.** Both functions SELECTRESIDUALGENERATORS and FINDCOMPUTATIONSEQUENCE were implemented in MATLAB. As computational tool, see [5], the algebraic equation solver MAPLE was utilized, which allows symbolic solving of algebraic loops. The input to the algorithm was the set of all 1058 MSO sets for the wind-turbine benchmark model, see Section 4.2, and the set of all 210 isolation classes for single fault isolation of all considered faults, see Sections 2.2 and 5.3.

```

(1): function FINDCOMPUTATIONSEQUENCE( $E', X'$ )
(2):    $\mathcal{C} := \emptyset$ 
(3):    $S := \text{FINDALLSCCs}(E', X')$ 
(4):   for  $i = 1, 2, \dots, |S|$  do
(5):      $(E_i, X_i) := S(i)$ 
(6):      $D_i := \text{DIFF}(X_i)$ 
(7):      $Z_i := \text{var}_D(E_i) \cap D_i$ 
(8):      $W_i := X_i \setminus \text{UNDIFF}(Z_i)$ 
(9):     if not ISINITCONDKNOWN( $Z_i$ ) then
(10):      return  $\emptyset$ 
(11):    end if
(12):     $E_{Z_i} := \text{GETDIFFERENTIALEQUATIONS}(E_i, Z_i)$ 
(13):     $E_{W_i} := E_i \setminus E_{Z_i}$ 
(14):     $S_{Z_i} := \text{FINDALLSCCs}(E_{Z_i}, Z_i)$ 
(15):    for  $j = 1, 2, \dots, |S_{Z_i}|$  do
(16):       $(E_{Z_i}^j, Z_i^j) := S_{Z_i}(j)$ 
(17):      if ISTOOLSOLVABLE( $Z_i^j, E_{Z_i}^j$ ) then
(18):         $\text{APPEND}(\mathcal{C}, (Z_i^j, E_{Z_i}^j))$ 
(19):      else
(20):        return  $\emptyset$ 
(21):      end if
(22):    end for
(23):    if ISJUSTDETERMINED( $E_{W_i}, W_i$ ) then
(24):       $S_{W_i} := \text{FINDALLSCCs}(E_{W_i}, W_i)$ 
(25):      for  $j = 1, 2, \dots, |S_{W_i}|$  do
(26):         $(E_{W_i}^j, W_i^j) := S_{W_i}(j)$ 
(27):        if ISTOOLSOLVABLE( $W_i^j, E_{W_i}^j$ ) then
(28):           $\text{APPEND}(\mathcal{C}, (W_i^j, E_{W_i}^j))$ 
(29):        else
(30):          return  $\emptyset$ 
(31):        end if
(32):      end for
(33):    else
(34):      return  $\emptyset$ 
(35):    end if
(36):  end for
(37):  return  $\mathcal{C}$ 
(38): end function

```

ALGORITHM 2: Find a computation sequence.

To investigate the sensitivity of SELECTRESIDUALGENERATORS to the parameter  $\gamma$ , that is, the tradeoff between properties 2 and 3 stated in Section 5.3 and reflected by  $|M|$  and  $\sum_{M \in \mathcal{S}} |M|$ , the algorithm was run with the wind turbine model and  $0 \leq \gamma \leq 1$ . The result is shown in Table 2, where  $\mathcal{S}$  denotes the set returned by SELECTRESIDUALGENERATORS. When  $\gamma = 1$ , the aim is to fulfill the isolation property with as few MSO sets as possible, no matter the size of the MSO sets. As seen in Table 2 this results in few, but large, MSO sets. The smaller the  $\gamma$ , the more attention is paid to the size of the MSO sets. It turns out that  $0.1 \leq \gamma \leq 0.6$  gives a decent tradeoff between  $|\mathcal{S}|$  and  $\sum_{M \in \mathcal{S}} |M|$  for the wind turbine model.

With  $\gamma = 0.5$ , the algorithm selected 16 MSO sets, that is,  $|\mathcal{S}| = 16$  and  $\sum_{M \in \mathcal{S}} |M| = 61$ . Of the 16 selected MSO sets, 7 contain algebraic equations only. The other 9 MSO sets contain both algebraic and differential equations.

TABLE 2: SELECTRESIDUALGENERATORS sensitivity to parameter  $\gamma$ .

$\gamma$	$ \mathcal{S} $	$\sum_{M \in \mathcal{S}}  M $
0.0	20	82
0.1	16	61
0.2	16	61
0.3	16	61
0.4	16	61
0.5	16	61
0.6	16	61
0.7	16	65
0.8	17	72
0.9	16	87
1.0	8	108

Thus, 7 of the 16 residual generators used in the final FDI system are static and the remaining 9 are dynamic. All 9 dynamic residual generators, due to the configuration of the algorithm, use integral causality. The *total* number of found residual generators is 34, that is,  $|\mathcal{G}| = 34$ , see Section 5.5. Of these 34 residual generators, 18 are static and the remaining 16 are dynamic.

**5.6.1. Fault Signature Matrix.** Given an MSO set  $M$ , its *fault signature*  $F(M)$ , with respect to the faults in  $F$ , is defined as

$$F(M) = \{f_i \in F : e_{f_i} \in M\}. \quad (17)$$

For instance, the fault signature of the MSO set  $M_1 = \{e_{26}, e_{27}\} \subseteq \mathcal{M}$  is  $F(M_1) = \{\Delta\omega_{g,m1}, \Delta\omega_{g,m2}\}$ . A convenient representation of the fault signature of a set of MSO sets  $\mathcal{S} = \{M_1, M_2, \dots, M_k\}$  with respect to  $F$  is the *fault signature matrix* (FSM)  $S$  with elements defined by

$$S_{ij} = \begin{cases} x, & \text{if } f_j \in F(M_i), M_i \in \mathcal{M}, \\ 0, & \text{else.} \end{cases} \quad (18)$$

The FSM for the 16 MSO sets on which the selected residual generators are based is given in Table 3.

## 6. Fault Detection and Isolation

For fault detection and isolation, diagnostic tests based on the output from each of the 16 residual generators are constructed. Since no assumptions are made regarding the nature of the faults that should be detected, see Section 2.2, nothing is known about the fault's temporal properties, size, rate of occurrence, and so forth. Hence, we may not be able to fully exploit the potential of some general method for change detection as, for example, the CUSUM test, see, for example, [16].

As said in Section 3, we, however, assume that no-fault training data is available. To take advantage of this fact and also handle uncertainties in terms of modeling errors and measurement noise, we base our diagnostic tests on a comparison of the estimated probability distributions of

no-fault and current residuals. The former probability distributions are estimated offline using the available no-fault training data and the latter online using current data. A clear advantage with this approach is that changes in mean and variance are handled in a unified way, since we consider the complete distribution of the residual.

**6.1. Diagnostic Test Design.** Let  $P^{\text{NF}}$  be a discrete estimate of the probability distribution of a residual from no-fault data, and  $P$  a discrete estimate of the distribution of the same residual from present data, both having  $n$  bins. Then, the Kullback-Leibler (K-L) divergence [17] between  $P$  and  $P^{\text{NF}}$  is given by

$$D(P \| P^{\text{NF}}) = \sum_{j=1}^n P(j) \log \frac{P(j)}{P^{\text{NF}}(j)}, \quad (19)$$

where  $P(j)$  denotes the  $i$ th bin of the discrete distribution  $P$ .

To apply the K-L divergence for construction of a diagnostic test, we proceed as follows. Given a representative batch of no-fault data  $\mathcal{Z}^{\text{NF}}$ , that is, in our case measurements of the variables in the set  $Z$  which contains the inputs and outputs to the model, we run the set of residual generators and obtain a set of residuals. For each residual  $r_i$ , we then estimate its probability distribution and obtain  $P_i^{\text{NF}}$ , that is, actually  $P_i^{\text{NF}} \approx P(R_i | \mathcal{Z}^{\text{NF}})$ , where  $R_i$  is a stochastic variable, discretized in  $n$  bins, representing residual  $r_i$ . As said, this procedure can be done offline. To estimate a probability distribution, we create a normalized histogram with  $n$  bins for the data from which the distribution should be estimated.

Online, we continuously estimate the distribution of the current residual  $r_i$  using a sliding window containing  $N$  samples of  $r_i$ . If we by  $P_i^t$  denote the estimated distribution of  $r_i$  calculated at time  $t$ , that is,  $P_i^t \approx P(R_i | \mathcal{Z}^t)$ , where  $\mathcal{Z}^t$  denotes the batch of data in the sliding window at time  $t$ , the diagnostic test is designed as

$$T_i(t) = \begin{cases} 1, & \text{if } D(P_i^t \| P_i^{\text{NF}}) \geq J_i, \\ 0, & \text{else,} \end{cases} \quad (20)$$

where  $J_i$  is the threshold for alarm. The K-L divergence  $D(P_i^t \| P_i^{\text{NF}})$  is referred to as the test quantity of the diagnostic test  $T_i$ .

**6.2. Fault Isolation Strategy.** Due to uncertainties not captured by the given model nor present in the no-fault training data, the power of diagnostic tests is not ideal for all faults. That is, the probability of detection given a certain fault is not always 1. To take this into account, the isolation scheme will interpret an “x” in a certain row in Table 3 as if the test *may* respond if the corresponding fault occurs and consequently no conclusions are drawn if a test does not respond, see [18].

To obtain the total diagnosis statement from a set of alarming diagnostic tests, we simply match their fault signatures with the FSM given in Table 3. For example, if only test  $T_{10}$  alarms, we look at the row corresponding to  $G_{10}$  and conclude that either fault  $\Delta\beta_1$  or  $\Delta\beta_{1,m2}$  are present. If then also  $T_{16}$  alarms, we combine the row corresponding to  $G_{16}$



TABLE 3: Fault signature matrix.

	$\Delta\beta_1$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\omega_g$	$\Delta\tau_g$	$\Delta\beta_{1,m1}$	$\Delta\beta_{1,m2}$	$\Delta\beta_{2,m1}$	$\Delta\beta_{2,m2}$	$\Delta\beta_{3,m1}$	$\Delta\beta_{3,m2}$	$\Delta\omega_{r,m1}$	$\Delta\omega_{r,m2}$	$\Delta\omega_{g,m1}$	$\Delta\omega_{g,m2}$
$G_1 (M_1)$														x	x
$G_2 (M_2)$												x	x		
$G_3 (M_3)$										x	x				
$G_4 (M_4)$								x	x						
$G_5 (M_5)$						x	x								
$G_6 (M_8)$					x										
$G_7 (M_{11})$				x									x		x
$G_8 (M_{27})$			x								x				
$G_9 (M_{29})$		x							x						
$G_{10} (M_{31})$	x						x								
$G_{11} (M_7)$														x	
$G_{12} (M_6)$															x
$G_{13} (M_{14})$				x								x		x	
$G_{14} (M_{28})$			x							x					
$G_{15} (M_{30})$		x						x							
$G_{16} (M_{32})$	x					x									

with the row corresponding to  $G_{10}$  and conclude that fault  $\Delta\beta_1$  must be present.

To handle also multiple faults, we use the fault signatures in the original FSM in Table 3 to create an extended FSM with fault signatures also for multiple faults. This is done by column-wise OR operations in the original FSM. For instance, the column in the FSM for the double fault  $\Delta\omega_{g,m1} \wedge \Delta\omega_{g,m2}$  will get “x” in rows corresponding to  $G_1$ ,  $G_7$ ,  $G_{11}$ ,  $G_{12}$ , and  $G_{13}$  and zeros elsewhere. In the fault isolation scheme, we first attempt to isolate all single faults using the original FSM in Table 3. If this does not succeed, we try to isolate double faults, and so forth.

## 7. Implementation Details

The final FDI system was implemented in SIMULINK according to the structure in Figure 2. The 16 residual generators were implemented as embedded MATLAB functions (EMF) in which the code was automatically generated from the structures obtained from the functions FINDCOMPUTATIONSEQUENCE and FINDRESIDUALGENERATORS. The initial conditions for the states in the dynamic residual generators were derived from the corresponding sensor measurements, if available, otherwise, set to zero. For instance,  $\theta_\Delta(t_0) = 0$ ,  $x_{\beta_{i1}}(t_0) = (\beta_{i,m1}(t_0) + \beta_{i,m2}(t_0))/2$ , and  $\omega_g(t_0) = (\omega_{g,m1}(t_0) + \omega_{g,m2}(t_0))/2$ . This may cause transients in the residuals, but this is not considered a problem.

**7.1. Parameter Discussion.** Although the aim is to keep the number of parameters in the automated design method at a minimum, there are nevertheless some parameters that must be set. This section lists the needed parameters and discusses their influence on the performance of the FDI system.

**7.1.1. Number of Histogram Bins and Size of Sliding Window.** The number of bins  $n$  in the histograms used as distribution estimates, is a tradeoff between detection time, noise sensitivity, and complexity, in terms of computational power and memory. A large  $n$  results in fast detection, but on the other hand also in increased sensitivity for noise. Also, a large  $n$  requires more memory and involves more computations, in comparison with a smaller  $n$ .

The size  $N$  of the sliding window used to batch data for creation of the histograms is a tradeoff between detection performance, noise sensitivity, and complexity. A large  $N$  will give the K-L test quantity lowpass characteristics, resulting in a smoothed K-L test quantity. This makes it possible to detect small changes in the estimated distributions. On the other hand, a large  $N$  requires more memory. The choice of  $N$  is also related to the number of bins  $n$  in the histograms and vice versa, since a small  $N$ , together with a large  $n$ , will result in a sparse histogram. Hence, the choices of  $N$  and  $n$  must match.

For the wind turbine benchmark model, investigations, however, indicate that the method is quite insensitive to the values of  $n$  and  $N$  if  $15 \leq n \leq 50$  and  $2000 \leq N \leq 6000$ . A decent tradeoff, taking this into account and also the complexity issues discussed above, is  $n = 20$  and  $N = 3000$ , which are the values used in the final FDI system.

**7.1.2. Alarm Thresholds.** The choice of alarm thresholds  $J_i, i = 1, 2, \dots, 16$ , is a tradeoff between detection time and the number of false detections. The higher the thresholds, the longer the detection time and the lower the rate of false alarms. The choice of alarm thresholds is related to the choices of  $n$  and  $N$  since both affect how sensitive a K-L test quantity is to noise, which in turn affects the rate of false

detections. We aim at choosing the alarm thresholds so that the number of false detections is minimized, implying that the choice of  $J_i$  must match the choices of  $n$  and  $N$ . For the wind turbine benchmark model, the alarm thresholds were computed as a safety factor  $\alpha = 1.1$  times the maximum value of the corresponding K-L test quantities from 100 simulations with no-fault data.

**7.1.3. Isolation Validation Time.** The only parameter involved in the fault isolation is the isolation validation time  $t_I^{\text{val}}$ . This parameter is used to compensate for the fact that the power of diagnostic tests is not ideal, see Section 6.2. This may, for example, result in that the detection times, for the same fault, are different for different diagnostic tests. To handle this, we demand that the output from the isolation has been equal for  $t_I^{\text{val}}$  samples before reporting the isolation result. By choosing a large  $t_I^{\text{val}}$ , we decrease the probability of false isolation, but on the other hand, increase the isolation time. For the wind turbine benchmark model, the isolation validation time  $t_I^{\text{val}}$  was set to 4 samples.

## 8. Evaluation and Results

To evaluate the performance of the proposed FDI system, we use the test cases described in [2]. The test cases are based on measured wind data and a sequence of injected faults. The set of injected faults, their time of occurrence and description, is specified in Table 4. The sequence contains 5 sensor faults and 3 actuator faults. Note that two faults are injected at 1000–1100 s, that is, at this time, we have the double fault  $\Delta\omega_{r,m2} \wedge \Delta\omega_{g,m2}$ .

The no-fault distributions used in the evaluation were estimated from residual data stemming from 100 Monte Carlo simulations with no-fault data, that is, inputs, corresponding to the measured variables in  $Z$ . Each set of no-fault data was generated with the provided wind turbine model with different noise realizations according to the model.

**8.1. Results and Analysis.** By means of Monte Carlo simulations, the FDI system was simulated 100 times with data from the provided wind turbine model setup according to the above-described test sequence.

Based on the results from the 100 runs, the mean time of detection  $\bar{T}_D$ , maximum time of detection  $T_D^{\text{max}}$ , minimum time of detection  $T_D^{\text{min}}$ , mean time of isolation  $\bar{T}_I$ , minimum time of isolation  $T_I^{\text{min}}$ , the total number of missed detections MD, and the total number of false detections FD, for each of the faults in the test sequence, were computed. The results along with the specified detection requirements [2], given in the row Req., are shown in Table 5, where all time values are given in seconds. Note that the specified requirements concern detection, and not isolation.

According to the row corresponding to  $T_D^{\text{max}}$  in Table 5, all faults in the test sequence could be detected. For faults  $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$ ,  $\Delta\beta_{1,m1}$ ,  $\Delta\beta_{3,m1}$ , detection requirements are met, by means of both  $\bar{T}_D$  and  $T_D^{\text{max}}$ .

All faults, except the double fault  $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$  could also be isolated. However, the mean time of isolation,  $\bar{T}_I$ ,

TABLE 4: Fault sequence.

Fault	Time (s)	Description
$\Delta\omega_{r,m2}$	1000–1100	$\omega_{r,m2} = 1.1\omega_{r,m2}$
$\Delta\omega_{g,m2}$	1000–1100	$\omega_{g,m2} = 0.9\omega_{g,m2}$
$\Delta\omega_{r,m1}$	1500–1600	$\omega_{r,m1} = 1.4 \text{ rad/s}$
$\Delta\beta_{1,m1}$	2000–2100	$\beta_{1,m1} = 5^\circ$
$\Delta\beta_{2,m2}$	2300–2400	$\beta_{2,m2} = 1.2\beta_{2,m2}$
$\Delta\beta_{3,m1}$	2600–2700	$\beta_{3,m1} = 10^\circ$
$\Delta\beta_2$	2900–3000	$\omega_n = \omega_{n2}, \zeta = \zeta_2$
$\Delta\beta_3$	3400–3500	$\omega_n = \omega_{n3}, \zeta = \zeta_3$
$\Delta\tau_g$	3800–3900	$\tau_g = \tau_g + 2000 \text{ Nm}$

for some faults, for example,  $\Delta\beta_{2,m2}$ , is substantially longer than the corresponding mean time of detection. The main reason for this is that some tests respond slower to faults than other. As said, fault  $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$  could not be isolated. In fact, this fault is not uniquely isolable with the isolation strategy described in Section 6.2 since the test response of fault  $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$  is a subset of the test response of fault  $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m1}$ , see Table 3. Both faults  $\Delta\omega_{g,m2}$  and  $\Delta\omega_{r,m2}$  are, however, contained in the diagnosis statement computed after the faults have been detected.

It seems like sensor faults, for example,  $\Delta\beta_{3,m1}$  tend to be easier to detect than actuator faults as, for example,  $\Delta\tau_g$  and  $\Delta\beta_2$ . One possible explanation may be that actuator faults in general cause changes in dynamics, whose effects are attenuated by modeling errors, noise, and so forth.

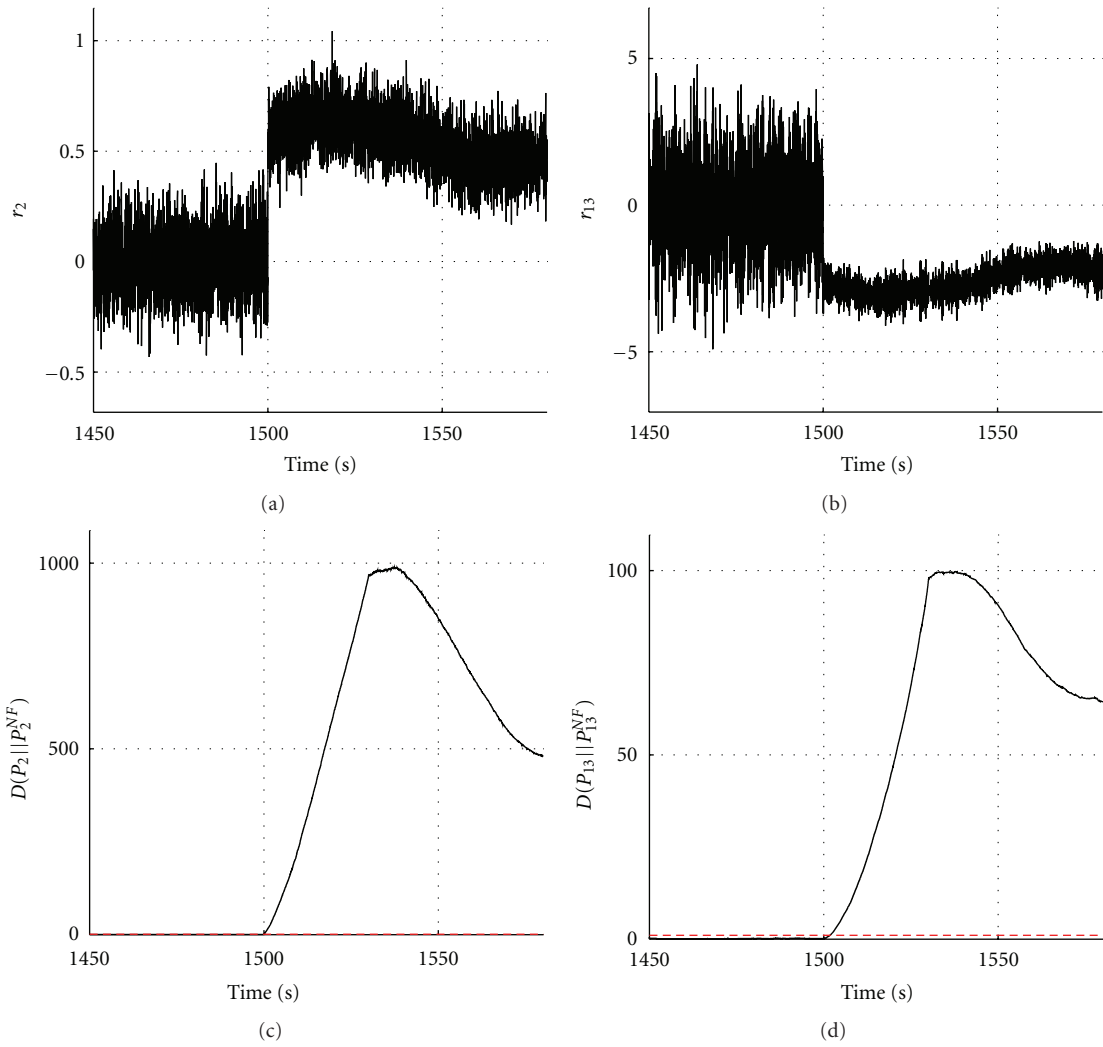
As can be seen in the last two rows of Table 5, there are no missed or false detections in any of the 100 test runs.

**8.2. Case Study of Fault  $\Delta\omega_{r,m1}$ .** To study in more detail how the FDI system handles faults, we consider the sensor fault  $\Delta\omega_{r,m1}$ . The fault corresponds to a fixed value of 1.4 rad/s being measured by sensor  $\omega_{r,m1}$  and occurs at time  $t = 1500$  s. According to the FSM in Table 3, the residuals sensitive to fault  $\Delta\omega_{r,m1}$  are  $r_2$  and  $r_{13}$ , obtained as output from the residual generators  $G_2$  and  $G_{13}$ , respectively. These residuals along with the corresponding K-L test quantities are shown in Figure 4. As can be seen, both the residuals and the test quantities respond distinctively to the fault.

To also illustrate the isolation procedure, we show in Figure 5 the result of the diagnostic tests  $T_2$  and  $T_{13}$  (a), the isolation result associated to faults  $\Delta\omega_{r,m1}$  (b) and  $\Delta\omega_{r,m2}$  (c), and also the signal that indicates when the isolation procedure is done (b, c). As can be seen in Figure 5, the first test that reacts to the fault is  $T_2$ . This occurs at  $t = 1500.23$  s. Since  $T_2$  is sensitive to both fault  $\Delta\omega_{r,m1}$  and  $\Delta\omega_{r,m2}$  and no other test has alarmed, the diagnosis statement is that either  $\Delta\omega_{r,m1}$  or  $\Delta\omega_{r,m2}$  may be present, and no fault can be isolated. At  $t = 1502.55$  s, test  $T_{13}$  alarms. Test  $T_{13}$  is sensitive to faults  $\Delta\omega_g$ ,  $\Delta\omega_{r,m1}$ , and  $\Delta\omega_{r,m2}$ , and the updated total diagnosis statement based on that both  $T_2$  and  $T_{13}$  have alarmed thus becomes  $\Delta\omega_{r,m1}$ , see Table 3. This occurs at time  $t = 1502.59$  s.

TABLE 5: FDI Results. Time values in seconds.

	$\Delta\omega_{r,m2}$ $\Delta\omega_{g,m2}$	$\Delta\omega_{r,m1}$	$\Delta\beta_{1,m1}$	$\Delta\beta_{2,m2}$	$\Delta\beta_{3,m1}$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\tau_g$
Req.	0.1	0.1	0.1	0.1	0.1	0.08	6	0.05
$\bar{T}_D$	0.040	0.16	0.058	4.30	0.069	51.57	18.1	7.94
$T_D^{\max}$	0.04	0.27	0.07	6.10	0.07	51.88	19.05	7.98
$T_D^{\min}$	0.03	0.06	0.05	0.40	0.06	50.57	16.37	7.90
$\bar{T}_I$	—	2.53	0.12	88.85	0.13	56.95	31.84	7.99
$T_I^{\max}$	—	3.13	0.12	114.26	0.13	120.73	111.96	8.03
$T_I^{\min}$	—	1.89	0.11	13.17	0.12	51.62	17.91	7.95
MD	0	0	0	0	0	0	0	0
FD	0	0	0	0	0	0	0	0

FIGURE 4: Affected residuals  $r_2$  (a) and  $r_{13}$  (b), and the corresponding K-L test quantities  $D(P_2^t || P_2^{NF})$  (c) and  $D(P_{13}^t || P_{13}^{NF})$  (d) at the time of occurrence of fault  $\Delta\omega_{r,m1}$ .

## 9. Conclusions

We have proposed an FDI system for the wind turbine benchmark designed by application of a generic automated design method, in which the numbers of required human decisions

and assumptions are minimized. No specific adaptation of the method for the wind turbine benchmark was needed. The method contains in essence three steps: generation of candidate residual generators; residual generator selection; diagnostic test construction. The second step is done by

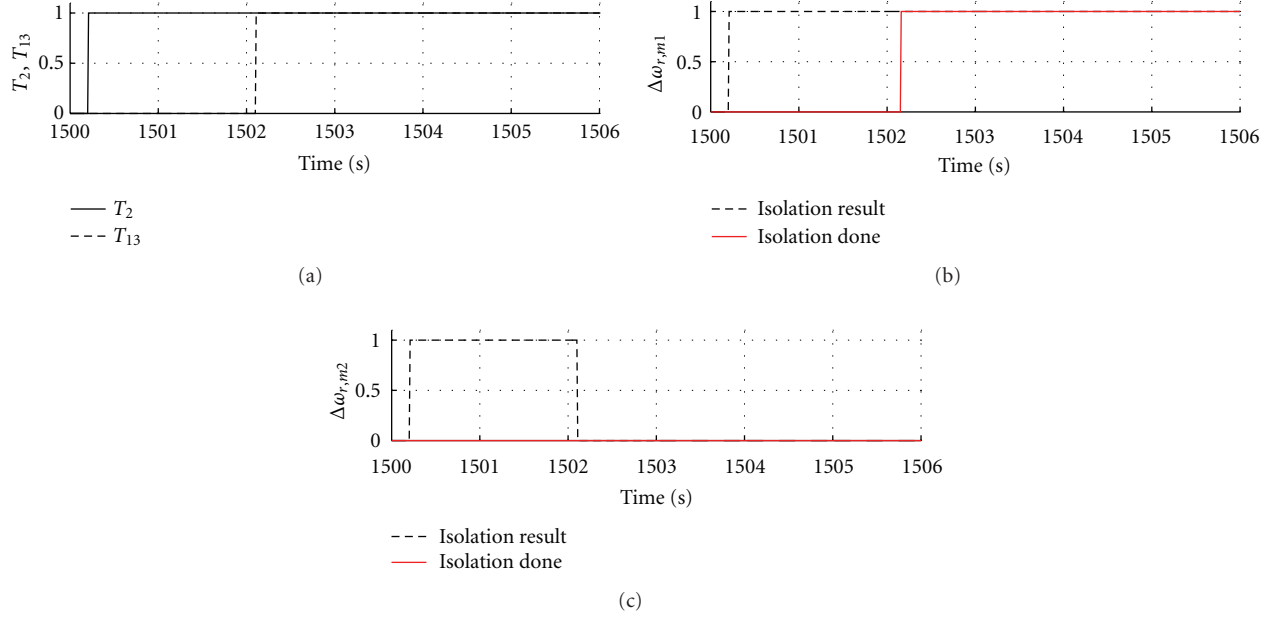


FIGURE 5: Isolation procedure for fault  $\Delta\omega_{r,m1}$ . (a) shows diagnostic tests  $T_2$  and  $T_{13}$ . (b and c) show the isolation result corresponding to faults  $\Delta\omega_{r,m1}$  and  $\Delta\omega_{r,m2}$ , respectively, and when the isolation procedure is done.

means of greedy selection, and the third step is based on a novel method utilizing the K-L divergence.

The performance of the proposed FDI system has been evaluated using the predefined test sequence for the wind turbine benchmark. The FDI system performs well; all faults in the test sequence were detected within feasible time and all faults, except a double fault, could be isolated shortly thereafter. In addition, there are no false or missed detections. A tailor-made, finely tuned, FDI system for the benchmark would probably perform better. However, in relation to the required design effort, and that no specific adaptation or tuning of the method to the benchmark was done, the performance is satisfactory.

## Appendix

### Algorithm for Finding a Computation Sequence

To make the paper more self-contained, the function `FINDCOMPUTATIONSEQUENCE` described in [5] is given as Algorithm 2. The function takes a just-determined equation set  $E' \subseteq E$  and a set of unknown variables  $X' \subseteq X$ , and it returns an ordered set  $\mathcal{C}$  as output. The algorithm assumes availability of a computational tool in the form of an algebraic equation (AE) solver such as, for example, Maple, see [5] for a thorough discussion regarding this. The function `FINDALLSCCs` is assumed to return an ordered set of equation and variable pairs, where each pair corresponds to a strongly connected component (SCC) of the structure of the equation set with respect to the variable set. There are efficient algorithms for finding SCCs in directed graphs, for example, the DM decomposition [19]. In MATLAB, the DM decomposition is implemented in the function `dmperm`.

Other functions used in `FINDCOMPUTATIONSEQUENCE` are as follows.

- (i) `DIFF` and `UNDIFF` takes a variable set as input and returns its differentiated and undifferentiated correspondence.
- (ii) `ISINITCONDKNOWN` determines if the initial conditions of the given variables are known and consistent, and the function `ISDIFFERENTIABLE` determines if the given variables can be differentiated with the available differentiation tool.
- (iii) `ISJUSTDETERMINED` is used to determine if the structure of the given equation set, with respect to the given variable set, is just determined. This is essential, since, otherwise, the computation of SCCs makes no sense.
- (iv) `GETDIFFERENTIALEQUATIONS` takes a set of equations and a set of differentiated variables as input and returns the differential equations in which the given differentiated variables are contained.
- (v) `ISTOOLSOLVABLE` determines if the available algebraic equation solver can solve the given equations for the given set of variables.
- (vi) `APPEND` takes an ordered set and an element as input and simply appends the element to the end of the set.
- (vii) The operator  $|\cdot|$ , taking a set as input, is assumed to return the number of elements in the set and the notion  $A(i)$  is used to refer to the  $i$ th element of the ordered set  $A$ .



## Acknowledgment

This work was supported by Scania CV AB, Södertälje, Sweden.

## References

- [1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, Springer, 2nd edition, 2006.
- [2] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault tolerant control of wind turbines âAS- a benchmark model," in *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pp. 155–160, Barcelona, Spain, 2009.
- [3] M. Krysander, J. Åslund, and M. Nyberg, "An efficient algorithm for finding minimal overconstrained sub-systems for model-based diagnosis," *IEEE Trans. on Systems, Man, and Cybernetics. Part A*, vol. 38, no. 1, pp. 197–206, 2008.
- [4] C. Svård, M. Nyberg, and E. Frisk, "A greedy approach for selection of residual generators," in *Proceedings of the 22nd International Workshop on Principles of Diagnosis (DX-11)*, Murnau, Germany, 2011.
- [5] C. Svård and M. Nyberg, "Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems," *IEEE Transactions on Systems, Man, and Cybernetics. Part A*, vol. 40, no. 6, pp. 1310–1328, 2010.
- [6] W. J. Rugh, *Linear System Theory*, Prentice Hall Information and System Sciences, chapter 13, 1996.
- [7] P. F. Odgaard, "Wind turbine benchmark model," 2011, <http://www.kkelectronic.com/Default.aspx?ID=9385>.
- [8] M. Staroswiecki and P. Declerck, "Analytical redundancy in non-linear interconnected systems by means of structural analysis," in *Proceedings of the IFAC Advanced Information Processing in Automatic Control*, (AIPAC'89), pp. 51–55, Nancy, France, 1989.
- [9] J. P. Cassar and M. Staroswiecki, "A structural approach for the design of failure detection and identification systems," in *Proceedings of the IFAC Control of Industrial Systems*, pp. 841–846, Belfort, France, 1997.
- [10] M. Staroswiecki, "Structural analysis for fault detection and isolation and for fault tolerant control," in *Fault Diagnosis and Fault Tolerant Control*, Encyclopedia of Life Support Systems, Eolss Publishers, 2002.
- [11] B. Pulido and C. Alonso-González, "Possible conflicts: a compilation technique for consistencybased diagnosis," *IEEE Trans. on Systems, Man, and Cybernetics. Part B*, vol. 34, no. 5, pp. 2192–2206, 2004.
- [12] S. Ploix, M. Désinde, and S. Touaf, "Automatic design of detection tests in complex dynamic systems," in *Proceedings of the 16th IFAC World Congress*, vol. 16, pp. 478–483, Prague, Czech Republic, 2005.
- [13] L. Travé-Massuyès, T. Escobet, and X. Olive, "Diagnosability analysis based on component-supported analytical redundancy relations," *IEEE Transactions on Systems, Man, and Cybernetics. Part A*, vol. 36, no. 6, pp. 1146–1160, 2006.
- [14] M. Krysander and E. Frisk, "Sensor placement for fault diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics. Part A*, vol. 38, no. 6, pp. 1398–1410, 2008.
- [15] M. R. Garey and D. S. Johnson, *Computers and Intractability—A Guide to the Theory of NPCompleteness*, W. H. Freeman and Company, 1979.
- [16] F. Gustafsson, *Adaptive Filtering and Change Detection*, Wiley, 2000.
- [17] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [18] M. Nyberg, "Automatic design of diagnosis systems with application to an automotive engine," *Control Engineering Practice*, vol. 7, no. 8, pp. 993–1005, 1999.
- [19] A. L. Dulmage and N. S. Mendelsohn, "Coverings of bi-partite graphs," *Canadian Journal of Mathematics*, vol. 10, pp. 517–534, 1958.

